CSE200: Computability and complexity
Beyond NP

Shachar Lovett
April 16, 2013

1 CoNP

The class NP corresponds to problems which can be verified efficiently. The class coNP is the opposite - problems which can be refuted efficiently. Formally, for a language $L \subseteq \{0, 1\}^*$ let $L^c = \{x \in \{0, 1\}^* : x \notin L\}$ be the complement. Then

$$coNP = \{L^c : L \in NP\}.$$ 

For example,

$$UNSAT = \{\text{formulas with no satisfying assignment}\}$$

is in coNP.

**Definition 1.1.** A language $L$ is coNP hard if for any $L'$ in coNP we have $L' <_p L$. If $L$ is in coNP then $L$ is said to be coNP complete.

**Theorem 1.2.** $UNSAT$ is coNP complete.

**Proof.** Fix some efficient encoding of CNF formulas in $\{0, 1\}^*$ which is onto. Then $UNSAT = SAT^c$. If $L$ is in coNP then $L^c \in NP$, hence $L^c <_p SAT$, hence $L <_p UNSAT$. \qed

Here is a different view on NP and coNP.

2 Polynomial hierarchy

Let $C$ be a family of languages (like $P$, $NP$, $EXP$). We define $\exists^p C$ to be the set of languages $L$ for which there exists $c > 0$ and a language $L' \in C$ such that

$$x \in L \iff \exists y, |y| \leq |x|^c, (x, y) \in L'.$$

Define $\forall^p C$ to be the set of languages $L$ for which there exists $c > 0$ and a language $L' \in C$ such that

$$x \in L \iff \forall y, |y| \leq |x|^c, (x, y) \in L'.$$
Lemma 2.1. $NP = \exists P$ and $coNP = \forall P$.

Proof. This is by definition. A language $L$ is in $NP$ if there exists a poly-time Turing machine $M$ such that
\[ x \in L \iff M(x, y) = 1. \]
Assume $M$ runs in time $n^c$. Consider the language $L' = \{(x, y) : M(x, y) = 1\}$. It is in $P$ since $M$ is poly-time computable. Hence
\[ x \in L \iff \exists y, |y| \leq |x|^c, (x, y) \in L'. \]
Similarly, if such an $L'$ exists we can define $M$ to be the Turing machine deciding $L'$.

This allows us to define a hierarchy of complexity classes. Define
\[ \Sigma_2 = \exists \forall P, \quad \Pi_2 = \forall \exists P \]
and generally
\[ \Sigma_i = \exists \Pi_{i-1}, \quad \Pi_i = \forall \Sigma_{i-1}. \]
For example, $L \in \Sigma_3$ if
\[ L = \{x : \exists y \forall z \exists w, M(x, y, z, w) = 1\} \]
where $M$ is poly-time computable.

We define the polynomial hierarchy as
\[ PH = \bigcup_{i \geq 1} \Sigma_i = \bigcup_{i \geq 1} \Pi_i. \]
Consider for example the problem of formula minimization.

$$MIN - CNF = \{(\varphi, 1^k) : \text{there is a CNF } \psi \text{ of size } \leq k \text{ computing } \varphi\}$$

Claim 2.2. $MIN-CNF$ is in $\Sigma_2$.

Proof. To verify that $(\varphi, 1^k) \in MINCNF$ we need to provide a CNF formula $\psi$ of size $\leq k$ such that $\psi \equiv \varphi$. To verify this we need to check all inputs. Hence
\[ (\phi, 1^k) \in MINCNF \iff \exists \psi(|\psi| \leq k)\forall x, \phi(x) = \psi(x). \]

3 Collapses

We believe that $P \neq NP$ and $NP \neq coNP$, and moreover that all the classes $\Sigma_i, \Pi_i$ are distinct. The following theorem gives some intuition for this.

Theorem 3.1. If $\Sigma_i = \Pi_i$ then $PH = \Sigma_i$. If $P = NP$ (or $P=coNP$) then $PH = P$. 

2
Proof. For simplicity, let’s assume NP = coNP and prove that this implies that \( \Sigma_2 = NP \). Let \( L \) be a language in \( \Sigma_2 \). Then

\[
L = \{ x : \exists y, |y| \leq n^c, \forall z, |z| \leq n^c, M(x, y, z) \}
\]

where \( M \) is poly-time computable. Define the language

\[
L' = \{ (x, y) : \forall z, |z| \leq n^c, M(x, y, z) \}.
\]

Clearly \( L' \) is in coNP. Since we assume \( NP = coNP \) we can find rewrite \( L' \) as

\[
L' = \{ (x, y) : \exists z, |z| \leq n^c, M'(x, y, z) \},
\]

where \( M' \) is also poly-time computable. Hence

\[
L = \{ x : \exists y, |y| \leq n^c, \exists z, |z| \leq n^c, M'(x, y, z) \}.
\]

Now we can ”combine” \( y, z \) to a single input, removing one level of alternation.  \qed